ARMY RESEARCH LABORATORY

Fuzzy-Logic Control of Battlefield Communications

Aivars Celmiņš

ARL-TR-1215

October 1996

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 1996

3. REPORT TYPE AND DATES COVERED
Final, August 1995–February 1996

4. TITLE AND SUBTITLE
Fuzzy-Logic Control of Battlefield Communications

5. FUNDING NUMBERS

6. AUTHOR(S)
Aivars Celmiņš

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-SC-CC
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1215

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The basic communications network in a battlefield is a single low-bandwidth radio channel that is used simultaneously by a small group of stations. In a modern Army environment, the nodes that share the radio channel will be computers that compose, encode, broadcast, receive, and display messages. Because the radio channel is used simultaneously by several nodes, one can expect collisions of messages, particularly during high combat activities. To reduce the number of such collisions, the access to the radio channel must be controlled. This report describes a control procedure that utilizes the computing power of the nodes and is based on fuzzy-logic rules. The control algorithm runs independently and concurrently in all participating nodes, making the control practically invulnerable. The control rules are designed to achieve and maintain high rates of information throughput, particularly under congested conditions. The control can be fielded using commercially available software or in-house developed programs.

14. SUBJECT TERMS
battlefield communications, distributed communications control, cooperative control, fuzzy-logic control

15. NUMBER OF PAGES
38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

INTENTIONALLY LEFT BLANK

Contents

INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK

1. Introduction

The subject of this report is the control of radio communications among combat units in a battlefield. The basic part of a battlefield communications network typically consists of a single low-bandwidth radio channel that is concurrently used by a moderate number of nodes. One expects that in a modern military environment the nodes will be computers that automatically prepare, encode, and transmit digitized messages. Hence, a battlefield communication network can be described as a single-channel radio connection of a moderate number of nodes that have high computing capacities. The information to be transmitted will be entered into the computer of a node, for instance, by keying or through automated sensors. The information that is received from the network will be stored in the computer and appropriately displayed on demand. The elimination of voice transmissions and human operators significantly improves the message transmission capacity of the network. All messages can be standardized according to a fixed protocol, and a much larger amount of information can be transmitted in a shorter time than with voice transmissions. An additional advantage of such a communication system is that the computers at the network nodes are available for operations other than data transmission. The excess computing capacity of the nodes can be used, for instance, for an analysis of the received data by an expert system that provides an evaluation of the tactical situation of the combat unit. On a more mundane level, the computers will be used to manage message queues in their memories and to keep records of broadcast times and acknowledgments for each message.

Communication problems in a digitized battlefield network might arise when the network is overloaded, for instance, during high combat activity when several nodes have queues of messages waiting to be transmitted. Without a regulated access to the broadcasting channel more than one node might try to access the channel at the same time thereby causing message collisions and a breakdown of communications. Therefore, access to the channel must be controlled. The present report is a description of an access control mechanism that takes advantage of the available computing power at the nodes. The control is autonomous and distributed (that is, the control algorithms run independently and concurrently in all network nodes), and the control algorithms are based on fuzzy logic. The motivation for this particular type of control is as follows.

A central hierarchical control that assigns access times to each node is not practical for two reasons. First, to be efficient, a central control needs current information about the state of all nodes in the network, but the network conditions are likely to change dynamically and unpredictably as the number of active nodes, their locations, and their message loads change. The acquisition of information about the states of the nodes would have to use some broadcasting time, and the obtained information might be outdated on arrival. Second, the delegation of the control to a single controlling node would increase the vulnerability of the communications, because the network would lose control when the controlling node were incapacitated. Therefore, we chose a *distributed control* system where the computer at each node autonomously controls its own access in such a manner that a high throughput rate of information is maintained.

For the design of an algorithm that controls at each node the access to the network the following was considered. The algorithm obviously should depend on such information about the current state of the network that is available to all nodes by passive listening to the network traffic, for instance, on the frequency of network accesses or on the number of message collisions. Since that information is only approximate and incomplete (for instance, it does not contain data about message queues at other nodes or about the states of nodes that are presently not broadcasting), the control algorithms must be able to accommodate approximate input. Second, the goal of the control is not formulated in terms of a set point but only approximately defined as "an increase of message throughput rate." Therefore, traditional control, such as PID, that depends on a set point and error term, would be difficult to implement. A method of choice in such situations is a fuzzy-logic control that can handle approximate inputs and open-ended control goals. Hence, the control method chosen and described in this report is a *distributed fuzzy-logic control.* The rules for the control algorithms were developed experimentally and tested on a computer model for battlefield networks (Celmiņš 1995). An efficient implementation of the fuzzy controller on PC-type computers at the nodes can be realized by using commercially available fuzzy-logic software and hardware.

We postulate three requirements for the access control algorithms:

(1) All nodes should have equal access possibilities.
(2) High priority messages should have preferred access.
(3) The channel should maintain high message throughput rates.

The usefulness of the second requirement is obvious but it is not trivial to implement because the nodes have no information about the priority levels of messages at other nodes.

Section 2 describes the general properties of the network and the network access procedure. Sections 3 and 4 describe the network monitoring algorithms, and the network access control rules, respectively. In Section 5, we present examples and Section 6 contains a summary and conclusions.

2. General Properties of the Network

2.1. Messages

For the management of radio communications, a "message" is merely a certain length of network time that is allocated for transmission. However, the management can be more efficient if several channel usage modes are distinguished, such as, the channel usage time for delivered messages, collided messages, not acknowledged messages, and messages corrupted by noise. This distinction is possible even for a passive observer of the network because the messages have a definite structure that is dictated by the message exchange protocol. A typical protocol is described in Kaste *et al.* (1992). In that protocol, each message consists of at least the first two of the following four parts: a head that contains an identification of the addressee and information about coding, a main message part containing the information to be transmitted, a pause during which the addressee is supposed to start an acknowledgment, and a tail that contains the acknowledgment. By listening to the net and examining the contents of the intercepted messages, an observer can estimate the mode of channel usage. For instance, if the tail of a message is missing, then it has not been acknowledged. If the head or the main body is corrupted, then either messages have collided or the channel contains noise. A further analysis of the corrupted message might clarify which is the case. And, of course, an extended silence indicates that the channel is idle and available for broadcasting.

In this report, we do not consider noisy channels. Therefore, the control rules are only concerned with delivered, not acknowledged, and collided messages, and idle time.

2.2. Network Access Management

By listening continuously to the net and analyzing the transmitted messages, a network access manager can determine at any time whether the radio channel is occupied with broadcasting, pausing after the broadcasting of the first two parts of a message, or is free. When the channel is found to be free and available for broadcasting at time t, then the network access manager module that is located in every node executes a simple access procedure, which is schematically shown in Figure 1.

First, the access manager checks the *message queue manager* module for messages to be broadcast. If there are messages in a message queue, then the queue manager identifies a message that should be broadcasted first and the access manager determines a broadcasting time t_b for that message. The broadcasting time is computed by randomly choosing a time from an interval $(t, t+D)$ with prescribed D. The size D of the interval is determined by the *access time controller* module (see Section 4), and the computation of D takes into account the current network conditions that are provided by a *network monitor* module (see Section 3). Next, the access manager continues to listen to the network, and if the network remains free up to the intended broadcasting time t_b, then the message is broadcast at that time. Otherwise, the procedure is aborted and the access manager waits

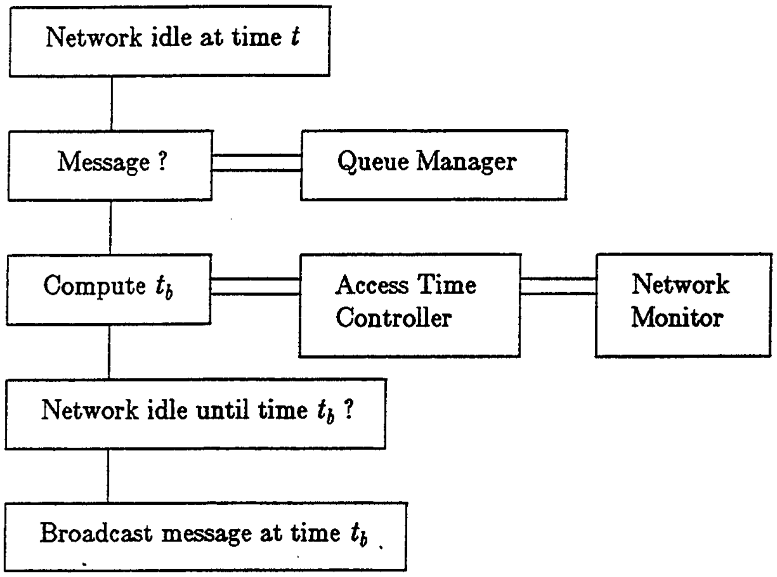

Figure 1. Schematic of the network access manager module.

for the next free network time to repeat the algorithm from the beginning.

Because all nodes continuously monitor the net, they will all determine at the same time t that the net is free and the access manager at each node with a queue will determine a set time t_b for its broadcasting. Since the set times are chosen randomly from a finite interval, the chances are small that any two set times are equal and there will be a smallest set time. The node with the smallest t_b will start broadcasting at that time, while all other nodes will find at their set times that a broadcasting already takes place, abort the access process, and wait for the next free time. Therefore, theoretically, this access procedure provides equal access chances for all nodes, and at the same time, avoids all message collisions.

Practically, however, the probability of message collisions is not zero because the propagation speed of radio signals is finite and there is a finite time delay within a computer between the determination that the channel is free at t_b and the actual broadcasting. Let the total time delay be α so that the time at which other nodes would recognize that a first broadcasting has already started and would be able to abandon their broadcast preparation is $t_b+\alpha$. Then, any node that has set a broadcasting time between the set time of the first node, t_b, and $t_b + \alpha$ will falsely determine that the channel is free at its set time, start broadcasting, and cause a message collision. The size of α has been estimated to about 0.5 s.

The described access procedure gives all nodes equal chances for accessing the channel if their *access time delay intervals* D from which the broadcasting times are chosen are equal. The computation of the intervals is governed by an access time control program that is identical for all nodes. Therefore, the access time intervals D will be equal among the nodes whenever the input information from the network monitors is the same for all nodes. We assume that this is the case for all active nodes that have monitored the network during a reasonable amount of time. Details about the monitoring are given in Section 3.

To provide high-priority messages with better chances for accessing the network, the network access manager uses shorter access time delay intervals for such messages. Let p be the priority of message (a number between 1 and 10) and D be the value of the access time delay interval that is provided by the controller (and is equal for all nodes). Then the access manager computes

$$D_{local} = D \cdot (1 - 0.09 \cdot p) \ , \qquad (2.1)$$

and chooses t_b from the interval $(t, t+D_{local})$. Hence, a node that has a high-priority message ready for broadcasting has a higher probability to select the smallest t_b and to start broadcasting than a node with a low-priority message.

The described procedure satisfies the first two requirements on the system that were postulated on page 2: equal access chances for all nodes and preferred access for high-priority messages. To achieve and maintain also a high information throughput rate, the number of message collisions as well as the idle time between messages must be kept at low levels. The probability of collisions decreases if t_b is chosen from an interval with large D. On the other hand, if D is large, then also the idling intervals between messages will be large in the average. Therefore, there exists an optimal size of D that corresponds to a maximum information throughput and that somehow depends on the status of the network. The task of the access time controller is to determine dynamically an optimal size of D from information that is obtained by passive monitoring of the network traffic. Section 3 describes the monitoring of the network, and Section 4 describes the control rules for the access parameter D.

2.3. Message Queues and Their Management

The task of the queue manager module in each node is to maintain for each message a record of submission time, message priority, times of unsuccessful broadcasts, and the time of a successful broadcast. In addition, the queue manager program must select one message from the queue that should be broadcast first when access to the broadcasting channel becomes available. The identification of a "first" message is the only interaction between the network access manager module and the queue manager module. The method of message selection has no bearing on the development and performance of the access control.

As an example of a possible message selection process, we present a description of a selection algorithm that was used in numerical experiments for the development of the access controller. The message was selected by computing a queuing weight q for each message in the queue and choosing the message with the largest weight for broadcasting. The message weight was computed from the following message properties:

- the time of message submission to the queue, t_o, [s];
- a priority index between one and ten, p;
- and the number of unsuccessful transmissions, N.

The formula for the queuing weight was

$$q = (1 + p + N) \cdot \max \left\{ 0.01 \, , \; \exp \left[- \left(\frac{t - t_o - 600}{600} \right)^2 \right] \right\} , \tag{2.2}$$

where the times are expressed in seconds. With this formula, the queuing weight q increases with the priority index p and the number N of unsuccessful repetitions of broadcasting. It also increases with the waiting time $t - t_o$ if the message has been less than 10 minutes in the queue. After 10 minutes, the message's queuing weight decreases again assuming that later submitted messages might supersede its contents.

3. Monitoring a Battlefield Network

3.1. Network State Parameters

The information about the status of the network that is obtained by listening to the network traffic is only approximate, but it has some other advantages: the acquisition of the information does not use network time, the information is up to date, and all nodes receive essentially the same information. In the present study, three groups of network state parameters that can be obtained by passive listening were considered. All parameters in these groups are time averages of observed quantities. The first group consists of the relative usages of channel access times in four usage categories during a time interval L prior to the reference time t. The second group consists of the relative numbers of network accesses during the same time interval and for the same categories of access types. The third group consists of the average message lengths during the same interval and for the same categories.

The four categories are as follows:

(1) Idle time.
(2) Successful (acknowledged) transmission.
(3) Collided transmission.
(4) Not acknowledged transmission (failure at the addressee).

3.2. Monitoring the State Parameters

The time averages of the network state parameters were computed over a monitoring and averaging interval of length L. That is, the computed average at time t represents the average value of the observed quantity during the time interval $(t - L\ ,\ t)$.

The formulas for the computation of the averages were as follows. Let the channel be occupied during the monitoring time by the usage category k at m_k time intervals of lengths $\Delta_i^{(k)}$ and with the mid-points $t_i^{(k)}$, $i = 1, \ldots, m_k$. Then the total time for that usage category is $\sum \Delta_i^{(k)}$ [s], and the relative usage time is

$$u_k(t) = \frac{1}{L} \sum_{i=1}^{m_k} \Delta_i^{(k)} \quad , \quad k = 1\,, \ldots, 4\ . \tag{3.1}$$

We also compute a trend indicator for each relative usage time with the help of a weighted averaging as follows. Let $w(\tau)$, $\tau \in [t - L\,,\,t]$ be a linear weight function defined by

$$w(\tau) = 1 - \frac{\tau - (t - L)}{L} \cdot w_\text{o} \quad , \tag{3.2}$$

where w_o is a parameter in [0,1]. Using this weight function, we compute by analogy with Eq. (3.1) a weighted relative usage time

$$\hat{u}_k(t) = \frac{1}{L \cdot w(t - L/2)} \sum_{i=1}^{m_k} \Delta_i^{(k)} \cdot w(t_i^{(k)}) \quad , \quad k = 1\,, \ldots, 4 \quad . \tag{3.3}$$

Because the weighted average $\hat{u}_k(t)$ is computed with larger weights for earlier parts of the averaging time interval, the difference

$$\check{u}_k(t) = u_k(t) - \hat{u}_k(t) \tag{3.4}$$

can be used as a trend indicator for the function $u_k(t)$. Numerical experiments with the network model BATNET (Celmiņš 1995) have shown that for control purposes the trend is appropriately characterized by $\check{u}_k(t)$ when w_o has a value of about 0.5.

The relative numbers of network accesses and the trends of the access numbers are computed only for the categories two, three, and four. In category No. 1, the number of idling "accesses", that is, the number of idle time intervals essentially equals the sum of all other accesses and does not carry additional information. The computation is done as follows. Let m_k, $k = 2{,}3{,}4$ be the number of accesses in the k-th category during the monitoring interval. Then the relative numbers of accesses are

$$n_k(t) = m_k \ / \sum_{i=2}^{4} m_i \quad , \qquad k = 2{,}3{,}4\ . \tag{3.5}$$

To compute the trend of $n_k(t)$, we again use the weight function $w(\tau)$ defined by Eq. (3.2). First, we obtain a weighted count by

$$\hat{m}_k = \sum_{i=2}^{m_k} w(t_i) \tag{3.6}$$

and a relative weighted count by

$$\hat{n}_k(t) = \hat{m}_k \;/\; \sum_{i=2}^{4} \hat{m}_i \ . \tag{3.7}$$

Then, the trend indicator $\check{n}(t)$ of $n_k(t)$ is computed by

$$\check{n}_k(t) = n_k(t) - \hat{n}_k(t) \ , \quad k = 2\,,3\,,4\,. \tag{3.8}$$

The average message length is computed by a simple averaging of the message intervals Δ_i for categories two, three, and four. Let $\Delta_i^{(k)}$, $i = 1, \ldots, m_k$, $k = 2,3,4$ be the interval lengths, and $t_i^{(k)}$ be the corresponding times of occurrence. Then the average message length in the category k is

$$a_k(t) = \sum_{i=1}^{m_k} \Delta_i^{(k)} \;/\; \sum_{r=2}^{4} m_r \ , \quad k = 2\,,3\,,4\,, \quad [\mathrm{s}]\ . \tag{3.9}$$

The weighted average length is

$$\hat{a}_k(t) = \sum_{i=1}^{m_k} \Delta_i^{(k)} \cdot w(t_i^{(k)}) \;/\; \sum_{r=2}^{4} \sum_{i=1}^{m_k} w(t_i^{(r)}) \ , \quad k = 2\,,3\,,4\,, \quad [\mathrm{s}]\ , \tag{3.10}$$

and the trend of the average length is computed by

$$\check{a}_k(t) = a_k(t) - \hat{a}_k(t)\ , \quad k = 2\,,3\,,4\,, \quad [\mathrm{s}]\ . \tag{3.11}$$

4. Controlling a Battlefield Network

4.1. Functions of the Control Module

4.1.1. Control of Monitoring

The principal task of the access time control module is to determine the access time delay interval D from input that is provided by the network monitoring module (see Section 2). However, to make the system efficient for different network configurations, the monitoring algorithms in the monitoring module (see Section 3.2) must be adapted accordingly. This adaptation is also governed by the control module.

The numerical results that are obtained by the network monitoring described in Section 3.2 depend on two parameters: the weight-function parameter w_o for trend calculations (Eq. (3.2)) and the length L of the monitoring interval.

It was determined by numerical experiments that adequate trend indicators are obtained with a value of about 0.5 for w_o; that value was used in all network control simulations.

The monitoring interval L should be adjusted in dependence of the average message length. If L is too short (of the order of an average message length), then the computed relative averages oscillate between near zero and near 100%. If L is too long, then the averages are not sensitive to recent changes of network conditions. In either case, the averages would be useless for the characterization of the status of the network. Numerical experiments with the computer model BATNET have shown that satisfactory results are obtained only when L is much longer than the average message length a, and that the optimal size of L is approximately given by the relation

$$L \approx 50 \cdot a \quad [\mathrm{s}] \ . \tag{4.1}$$

In a presently used communications protocol (Kaste *et al.* 1992) a typical message length is about one to ten seconds. Therefore, if that protocol is used, then the optimal length of L is about one to ten minutes. Such a monitoring time is also reasonable with respect to military operations. If communication characteristics change due to troop operations then such changes are likely to become effective within several minutes or over a longer time. Compliance with the relation (4.1) is ensured by adjusting L dynamically. The corresponding control rules are outlined in Section 4.2.1.

4.1.2. Network Access Control

The traffic of messages through the broadcasting channel depends on the message broadcasting times t_b that are calculated by the network access management modules as described in Section 2.2. That computation is governed by the size of the access time delay interval D. Therefore, D is the principal parameter for access control. Its size is constantly updated in each node by the access time control module such that in the average the unwanted categories of network usage (idling and collision) are reduced. Input to the control algorithm consists of the monitored network state parameters described in Section 3.2., and its output is an update of D. The control rules for the computation of D are described in Section 4.2.2., and the fuzzy-logic implementation of the control rules is described in Section 4.3.

4.2. Control Rules

4.2.1. Rules for Monitoring

The length of the monitoring interval L should be about 50 times the average length a of a message. We express this requirement in terms of the quantity

$$Q = L \,/\, (50 \cdot a) - 1 \tag{4.3}$$

and adjust L such that Q is approximately zero. The adjustments are done with the aid of an adjustment factor $1 + \lambda$. Let L_{old} be the present value of the monitoring interval L and let L_{new} be its new (improved) value. Then the adjustment formula is

$$L_{new} = L_{old} \cdot (1 + \lambda) \ . \tag{4.4}$$

The value of λ is determined from the present value of Q by the following rules:

If $\quad Q$ is (negative, zero, positive)
then $\quad \lambda$ is (positive, zero, negative).

We express these rules in the form of a rule table:

Q	NL	N	Z	P	PL
λ	PL	P	Z	N	NL

In this table, NL, N, Z, P, and PL stand for "negative large," "negative," "zero," "positive," and "positive large," respectively. In Section 4.3.2, the rules are quantified in terms of fuzzy sets. We note here only that the dependence of λ on Q should be weak in the sense that λ should change from zero to non-zero only when Q significantly deviates from zero.

4.2.2. Rules for Network Access

Access to the network is controlled by adjusting the size of the global access time delay interval D according to the status of the network. In contrast to the desired slow response, when controlling the monitoring interval L, the value of D should be adjusted in quick response to changing conditions of network traffic. To accelerate the response of the control, in addition to using the network state parameters as control input we also use their trends.

Rules for the control of access can be derived by the following considerations. An increase of D causes the relative idling time u_{idle} to increase and the relative collision time u_{col} to decrease, and vice versa. Therefore, a possible strategy for reducing both of these quantities is to choose D such that u_{idle} and u_{col} are approximately equal. This rule was found to be very effective, although fine-tuning of the rule showed that a better performance can be achieved when the relative collision time is kept smaller than the relative idle time. One must also consider the number and the lengths of colliding messages. A single collision involving a very long message can increase the average collision time u_{col} as much as many collisions involving short messages, but in each of these cases, the proper control strategy is different. Therefore, in a second rule, the trend of the number of colliding messages was used as input. By that rule, D is increased if the trend increases and vice versa.

To make the control more responsive to extreme conditions, two more inputs were considered: the idle time over a threshold and the collision time over a threshold. If these quantities exceed prescribed thresholds, then corresponding corrections of D are initiated.

These adjustment rules are implemented in principle by the following algorithm that runs independently and concurrently in each node. Let D_{old} [s] be the present value of D and δ be a corrector supplied by the control algorithm. Then the updated value D_{new} of D

is computed by the formula

$$D_{new} = D_{old} \cdot (1 + \delta) \ . \tag{4.5}$$

For this algorithm, the input for the control rules consists of the averaged network state parameters and their trends (see Section 3), and the output of the control rules is the value of the corrector δ.

In practice, the simple adjustment procedure (4.5) must be modified to ensure that after an initialization time the controllers in all nodes indeed produce approximately the same value D_{new}, including those nodes that join the network at different times and start the control algorithm with different initial values of D. For clarity, we first describe the control rules in terms of Eq. (4.5) and discuss the modifications of the algorithm later.

We arrange the control rules in the form of qualitative rule tables. In Section 4.3.2, we will make the rules more precise by quantifying them in terms of fuzzy sets. The rule tables are as follows.

Rule No. 1.
Input: Collision time minus idle time

$u_{col} - u_{idle}$	NL	N	Z	P	PL
δ	NL	N	Z	P	PL

Rule No. 2.
Input: Trend of the number of colliding accesses

$\check{n}_{col}$	NL	N	Z	P	PL
δ	NL	N	Z	P	PL

Rule No. 3.
Input: Idle time over a threshold T_{idle}

$u_{idle} - T_{idle}$	NL	N	Z	P	PL
δ	O	O	O	NL	NL

Rule No. 4.
Input: Collision time over a threshold T_{col}

$u_{col} - T_{col}$	NL	N	Z	P	PL
δ	O	O	O	PL	PL

In the last two rule tables, O denotes "no output" (that is, no rule is fired in these cases). The purpose of the last two sets of rules is to provide an accelerated change of D in extreme cases. Experiments show, however, that the last two sets of rules have only a minor effect on the performance of the control if used in addition to the first two sets of rules. When used without the first two sets of rules the performance of the control was not as good as with the first two rule sets alone.

We now discuss the modifications of Eq. (4.5) to achieve equal results among nodes that start with different initial conditions. To that end, we separate in Eq. (4.5) the dimensional factor D_{old} from the nondimensional correction factor $F=1+\delta$ and devise a different adjustment procedure for each factor.

For F we construct an adjustment procedure such that the factor drifts with repeated updating to a fixed value that is independent of the initial value of F. Let Δt [s] be the difference between the current and previous time of network parameter updates. (Average values of the network parameters are calculated at discrete times, namely, at the end of each message transmission period.) Let f be a fixed value of F to which the correction factor should drift with increasing time. Let $\epsilon = \exp(-\Delta t/60)$. Then the factor is updated as follows

$$F_{new} = (f \cdot (1-\epsilon) + F_{old} \cdot \epsilon) \cdot (1+\delta) \ . \qquad (4.6)$$

One can show that the exponential factor in the formula has the effect that after a few minutes of operation the computed value of F_{new} approaches $f \cdot (1+\delta)$ independently of the initial value of F. Therefore, if all nodes would use the same f, then within a few minutes of control operation, the effective value of F_{new} will be the same for all nodes. Experiments show, however, that it is not possible to assign *a priori* a fixed value to f that is appropriate for all network conditions. The effective value of the factor F typically must be allowed to vary between 0.001 and 80, and f should be dynamically assigned values in the same range. We, therefore, devise an algorithm that makes all nodes to change the value of f in lockstep within that range. The algorithm consists of assigning to f a finite set of discrete values within the range of interest and updating the present value f_{old} according to the following rules:

$$\begin{aligned} &\text{If } \delta_{old} > 0 \text{ and } \delta_{new} > 0 \quad \text{then} \quad f_{new} = \min\{ f_{old} + 4 \ , \ 80 \} \ ; \\ &\text{If } \delta_{old} < 0 \text{ and } \delta_{new} < 0 \quad \text{then} \quad f_{new} = \max\{ f_{old} - 4 \ , \ 0.001 \} \ . \end{aligned} \qquad (4.7)$$

This algorithm increases or decreases the value of f in steps of four whenever the corrector

δ continuously increases or decreases, respectively. After at most 20 corrections with the same sign, the bounds of f are reached and subsequent updates are done by all nodes in lockstep. Because in a high-traffic situation the correction factor δ typically increases or decreases continuously over more than 20 message transmissions, the equality of f among the nodes is established in a short time.

A reasonable value for the dimensional factor D_{old} is, for instance, a multiple of the average idle time interval i. In a network with n nodes, where all nodes use the same access time delay interval D the expected value of i is $D/(n+1)$ and the current value of D could be estimated from observations of i. However, the computed average value of i oscillates heavily even when the monitoring interval L is large because D is changed dynamically by the controllers and furthermore the idle intervals $\Delta_1 = t_b - t$ are computed by using the local delay intervals D_{local} instead of the global D (see Eq. 2.1). The local D_{local} depend on the priorities of the broadcasted messages. (If the priorities were assigned randomly, then the value of i would be about $D/(2(n+1))$, but random assignment of the priority index p cannot be assumed.) The oscillations of the average idle time interval i can be suppressed by averaging the last two computations of i and using an exponential weight factor that reduces the influence of the previous reading if it was made a long time ago. Let i_{prev} be the previous average idle time and i_{pres} be the presently computed value. Let $v = \exp(-\Delta t/300)$. Then a new value i_{new} of the average idle time interval is computed by

$$i_{new} = (i_{prev} \cdot v + i_{pres}) / (v + 1) \quad . \tag{4.8}$$

One may now set in Eq. (4.5) D_{old} equal to a multiple of i_{new} assuming that such a value approximates the global interval D that is presently used in the net. This creates, however, another problem. When D is increased, then the value of the average idle time i_{new} which is used for the next updating usually also increases and vice versa. Therefore, if D_{old} is always set equal to a multiple of the idle time i_{new}, then, at the next updating of D, the correction by the nondimensional correction factor F is, in general, amplified. Under certain conditions, this amplification can cause a drift of D to zero or infinity. A drift to zero can be avoided by setting for D_{old} a fixed lower bound for which 0.01 s was found to be adequate. An excessive increase of D_{old} is avoided by making it a function of the logarithm of the idle interval i_{new} (instead of a multiple of i_{new}). After some experimentation, the following final formula for the access delay time interval was chosen:

$$D_{new} = D_{old} \cdot F_{new} = (\, 0.01 + \log(1 + i_{new}/4\,)\,) \cdot F_{new} \quad , \quad [\,s\,] \, , \tag{4.9}$$

where i_{new} is expressed in seconds, and F_{new} is given by Eq. (4.6).

To summarize, the fuzzy-logic control rules provide a value of the corrector δ. That corrector and the average length i of the idling interval are used in Eqs. (4.6), (4.7), (4.8), and (4.9) to compute a new value D_{new} of the access time delay interval. For high-priority messages, it might be further modified as described in Section 2.2. (see Eq. (2.1), page 5).

4.3. Fuzzy-Logic Implementation of Control

4.3.1. The Compositional Rule of Inference and Defuzzification

The control rules for network monitoring and network access are outlined in Section 4.2. They all have the simple form

$$\text{if } x = A \text{ then } y = B \ , \tag{4.10}$$

where x is an observed network state parameter (for instance, Q or $\check{n}_{col}$), and y is a control parameter (λ or δ in our case). For the present application, the rules must be formulated such that they can be used in situations where the input x is known only approximately. That is, the classical inference rule *modus ponens*

$$\begin{array}{lcccc} \mathit{P1} & \text{if} & x = A & \text{then} & y = B \\ \mathit{P2} & & x = A & & \\ \hline C & & & \text{therefore} & y = B \end{array} \tag{4.11}$$

must be modified because modus ponens produces a conclusion only when x exactly equals A. If x is not exactly equal to A, then the rule provides no information about y.

We now present a modification of the modus ponens that allows approximate descriptions of the attributes of x and y and is commonly used in applications of fuzzy logic. Let $\widetilde{A}$, $\widetilde{B}$, $\widetilde{D}$, and $\widetilde{E}$ be fuzzy sets. Then the rule (4.11) is replaced by the following rule with fuzzy premises and a fuzzy conclusion:

$$\begin{array}{lcccc} \mathit{FP1} & \text{if} & x = \widetilde{A} & \text{then} & y = \widetilde{B} \\ \mathit{FP2} & & x = \widetilde{D} & & \\ \hline \mathit{FC} & & & \text{therefore} & y = \widetilde{E} \end{array} \tag{4.12}$$

In this rule, $\widetilde{A}$, $\widetilde{B}$, and $\widetilde{D}$ are given and the fuzzy conclusion FC is an algorithm for the computation of $\widetilde{E}$ from the three given fuzzy sets. One such algorithm that is commonly used in control problems is the *compositional rule of inference*, first formulated by Zadeh (1975). Similar other algorithms have also been designed for particular problems, but the original compositional rule of inference is most popular because of its simplicity and appeal to common sense. (See Klir and Yuan (1995), Pedrycz (1992), and Terano *et al.* (1991)).

We note in passing that typical compositional rules of inference are not strict generalizations of the modus ponens because they do not reduce to the latter if $\widetilde{A}$, $\widetilde{B}$, and $\widetilde{D}$ are crisp. A fuzzy-logic rule with that property is called a *generalized modus ponens.*

The premise *P1* of the modus ponens and the fuzzy premise *FP1* of Eq. (4.12) both define functional relations between elements of antecedent and consequent spaces. These relations have a simple geometric interpretation. Let the antecedent space be called u-space and the consequent space be called v-space. For illustration purposes let the spaces be one-dimensional. (The following considerations are valid in arbitrary dimensions, but it is simpler to present them in one-dimensional spaces.) Then A is a point on the u-axis, B is a point on the v-axis, and the *crisp premise P1*, that is, "A implies B" or "$A \rightarrow B$" can be interpreted as the definition of a function $y(x)$ that consists of a single point (A,B) in the u,v-plane. If the antecedent x has the value A, then the consequent y has the value B. For any other values of x, the premise does not specify a corresponding value of the consequent.

In the *fuzzy premise FP1,* the function $y(x)$ is generalized by replacing the crisp point (A,B) with a fuzzy set $\widetilde{R}_{A\rightarrow B}$ that is defined by a membership function $\mu_R(u,v)$ in the u,v-plane. In analogy to the crisp single-point function the projections of μ_R on the u- and v-axis are assumed to be the fuzzy sets $\widetilde{A}$ and $\widetilde{B}$, respectively, that are given in *FP1.* Figures 2 and 3 illustrate such a fuzzy relation. It is easy to see that this representation of *FP1* reduces to the crisp *P1* when the membership function μ_R degenerates into the characteristic function of a crisp point.

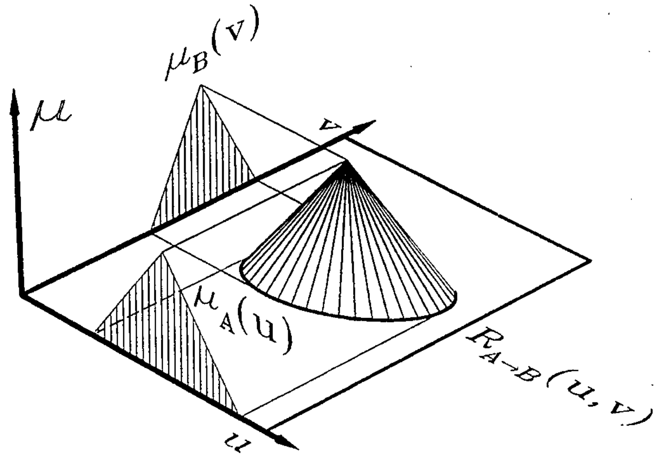

Figure 2. Fuzzy relation with a conical membership function.

The second fuzzy premise *FP2* states that we are given on the u-axis a fuzzy set $\widetilde{D}$ with a membership function $\mu_D(u)$ that may or may not be equal to the projection $\mu_A(u)$ of the fuzzy relation $\widetilde{R}_{A\rightarrow B}$. The algorithm of the fuzzy conclusion *FC* obviously should be such that when $x = \widetilde{A}$ then the resulting $\widetilde{E} = \widetilde{B}$. In addition, it should produce *some* $y(x)$ for any other input x. An algorithm with these properties can be obtained by a combination of geometrical intersections and projections of membership functions. It is illustrated in Figure 4. First, the membership function μ_R of the relation $\widetilde{R}_{A\rightarrow B}$ is intersected with the membership function μ_D of the input $\widetilde{D}$. In Figure 4, the intersection has the form of a shaded roof. Next, the intersection is projected onto the v-axis, and the projection defined as the membership function $\mu_E(v)$ of the fuzzy conclusion $\widetilde{E}$. In Figure 4, the projection $\mu_E(v)$ of the roof is a trapezoid in the v, μ-plane.

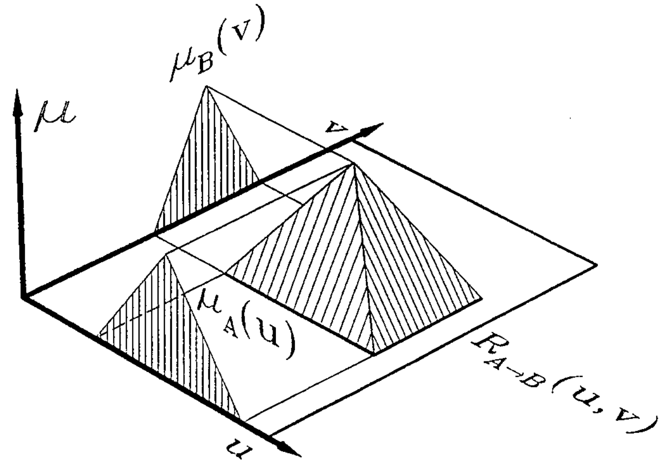

Figure 3. Fuzzy relation with a pyramidal membership function.

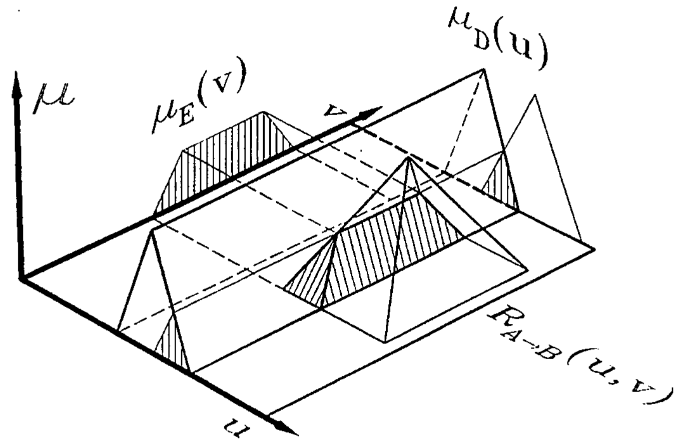

Figure 4. Compositional rule of inference.

It should be obvious that the conclusion set $\widetilde{E}$ depends on the form of the membership function $\mu_R(u,v)$. If, for instance, in Figure 4 that membership function would be a cone instead of a pyramid, then the intersection and its projection would be different from those in the figure. Therefore, the formulation of the fuzzy premise *FP1* in Eq. (4.12) is incomplete: one must specify in addition to (or instead of) the sets $\widetilde{A}$ and $\widetilde{B}$ also the set $\widetilde{R}_{A\rightarrow B}$. Moreover, also the concepts of intersection and projection must be precisely defined because one does not need to use the simple geometrical constructions shown in the example.

In most control problems it is assumed that the relational membership function $\mu_R(u,v)$ has the simple pyramidal form shown in Figures 3 and 4. The pyramid has the advantage that it can be easily constructed from given $\widetilde{A}$ and $\widetilde{B}$ and that arithmetical operations with such a membership function are simple. Methods for the construction of other types of relational membership functions are given in Terano *et al.* (1991). In our battlefield communications control, we use relational membership functions of pyramidal form.

For the concepts of intersection and projection, we follow Zadeh (1975) and use min and max operators, respectively. (These operations were also used in the illustrations in Figures 2, 3, and 4.) We construct the pyramidal membership function $\mu_R(u,v)$ by intersecting the membership functions of $\widetilde{A}$ and $\widetilde{B}$ and using the min operator for the intersection (see Figure 3):

$$\mu_R(u,v) = \min \{ \mu_A(u) \, , \, \mu_B(v) \} \ . \tag{4.13}$$

By projecting the pyramid onto the u- and v-axes with the max operator we recover the membership functions of $\widetilde{A}$ and $\widetilde{B}$, respectively:

$$\begin{aligned} \mu_A(u) &= \sup_v \mu_R(u,v) \ , \\ \mu_B(v) &= \sup_u \mu_R(u,v) \ . \end{aligned} \tag{4.14}$$

Next, we construct the fuzzy conclusion as shown in Figure 4. Let the input $\widetilde{D}$ in Eq. (4.12) be given by the membership function $\mu_D(u)$. We obtain the corresponding $\mu_E(v)$ of the conclusion $\widetilde{E}$ by intersecting $\mu_R(u,v)$ with $\mu_D(u)$ and projecting the result onto the v-space. The intersection is

$$\mu_I(u,v) = \min\{ \mu_R(u,v) \, , \, \mu_D(u) \} \ . \tag{4.15}$$

Its projection onto the v-space is

$$\mu_E(v) = \sup_u \mu_I(u,v) = \sup_u \min \{ \mu_R(u,v) \, , \, \mu_D(u) \} \ . \tag{4.16}$$

Usually, the inference rule is illustrated as shown in Figure 5 where the left-hand picture shows the view of the roof-like membership function as a shaded triangle in the u, μ-plane. The right-hand picture shows the view of the roof in the v, μ-plane where it has the form of a trapezoid. The figure is somewhat misleading, because it only applies to cases where μ_R is a pyramid with sides of its base parallel to the coordinate axes. If, for instance, μ_R were a cone then the inferred membership function μ_E would be a quadric similar to the dotted curve instead of the trapezoid. (The quadric is the outline of the intersection of a side of the prism μ_D with the cone μ_R.) The figure is, however, adequate in our case because we use in the control algorithms the pyramid computed by Eq. (4.13) and conclusions that are computed by Eq. (4.16).

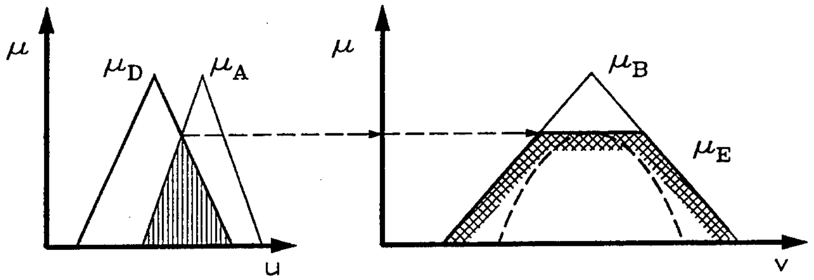

Figure 5. Inference rule in pyramid case.

If the control has only one rule, then only Eq. (4.16) is needed to obtain the fuzzy value of the control parameter. In praxis this is seldom the case. For example, the control of the monitoring length L in Section 4.2.1 has a single input Q and a single output λ, but the size of λ is governed by a rule that consists of five elementary rules listed in a rule table (see page 10). When the control is activated then in general more than one of these elementary rules will fire and the outcomes must be combined. Similarly, the four rules for network access control in Section 4.2.2 each consists of a set of five elementary rules in the rule tables (see pages 11-12). Again, several of the 20 elementary rules might fire simultaneously.

The combination of the outcomes of fuzzy rules is done by a fuzzy-logic OR operation. The outcome of each rule is a fuzzy number represented by a membership function in the consequent space. Let the outcomes of, say, three rules for δ have the membership functions $\mu_{\delta i}(v)$, $i=1,2,3$. In fuzzy logic, the OR operation corresponds to a union (maximum) of the individual functions, and the membership function μ_δ of the combined conclusion is calculated with the formula

$$\mu_\delta(v) = \max \{ \mu_{\delta 1}(v) , \mu_{\delta 2}(v) , \mu_{\delta 3}(v) \} . \tag{4.17}$$

Figure 6 illustrates such a combination of results from three different fuzzy rules. The combined curve is the membership function of the fuzzy output value of the control parameter.

Another method for the combination of rules is the so-called rule table method. This method is often used when rules are more complicated than (4.10), for instance, when they have the form

$$\text{if } x=A \text{ and } y=B \text{ then } z=C . \tag{4.18}$$

Using the rule table method, this combination is expressed in form of a look-up table, but similar look-up tables can be used also for the combination of simpler rules. For instance, a combination of the access control Rules No. 1 and No. 2 (Section 4.2.2) could be

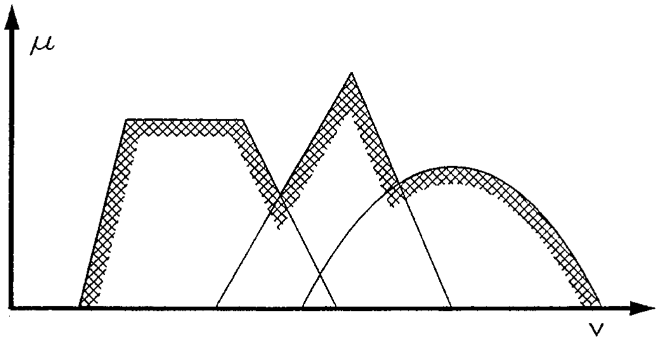

Figure 6. Union of consequences.

expressed by the following table with two entries:

Combined Rules No. 1 and No. 2.

		$u_{col} - u_{idle}$				
		NL	N	Z	P	PL
$\check{n}_{col}$	NL	NL	NL	N	N	Z
	N	NL	N	N	Z	P
	Z	N	N	Z	P	P
	P	N	Z	P	P	PL
	PL	Z	P	P	PL	PL

In general, the output from a union of rules is smoother than the output from rule tables. In experiments with the control of battlefield communications, the performance of the control was not consistently different when using rule tables or the union of the results of rules, respectively, and the differences were not significant. Therefore, a final decision to use the one or the other approach can be based on the effectiveness (speed and costs) of the hardware and software that implements the control in a fielded equipment.

The outputs of the fuzzy control algorithms are fuzzy values of the control parameters λ and δ. To activate the control, that is, to compute L_{new} and D_{new} by the formulas (4.4) and (4.9), respectively, the control parameters λ and δ must be given crisp values. The extraction of a crisp value from the result of a fuzzy calculation is called defuzzification. Many different defuzzification algorithms have been proposed and used for different applications. In the present application, we use the "center of gravity" method. By this method, the crisp output of the control rules is defined as the center of gravity of the fuzzy output $\mu(v)$. Thus, the center of gravity of the output $\mu_\delta(v)$ (see, e.g., Figure 6) is computed by the well-known formula

$$\delta = \int \mu_\delta(v)\ v\ dv \ / \int \mu_\delta(v)\ dv\ . \qquad (4.19)$$

A corresponding formula is used for λ.

4.3.2. Granulation of the Control Rules

To implement the control rules described in Section 4.3 on a computer, the input and output spaces (the antecedent and consequent spaces) must be "granulated." For instance, to implement the rule for λ, Section 4.3.1, one must define membership functions for the concepts "negative large," "negative," "zero," "positive," and "positive large" for Q and λ. In this case, a "zero Q" should cover a large interval around zero, because we want to change the monitoring control parameter L only when Q significantly deviates from zero. The choice of the granulation (that is, of the membership functions of the linguistic categories in antecedent and consequent spaces) determines the numerical effects of the control. The granulations presented in this report were obtained empirically by extensive runs of simulated battlefield communication sessions. For these simulations, the battlefield communications model BATNET (Celmiņš, 1995) was used. The granulation was modified until favorable information throughput rates were obtained for a large number of different network traffic conditions.

Figure 7 shows the membership functions of the five linguistic categories of Q, and Figure 8 shows the membership functions of the linguistic categories of the consequent λ.

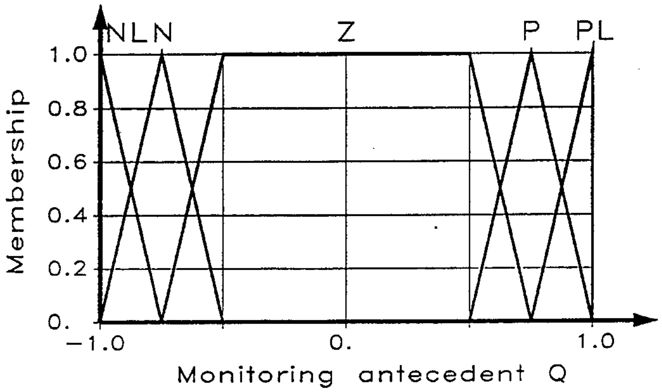

Figure 7. Granulation of the monitoring control antecedent Q.

The membership functions of the linguistic categories of Rules No. 1, 2, 3, and 4 (pages 11-12) are shown in Figures 9, 10, 11, and 12, respectively, and the membership functions of the consequent δ are shown in Figure 13. We notice slight asymmetries in the granulations of Rules No. 1 and 2 (Figures 9 and 10). These asymmetries were not chosen arbitrarily but developed by the tuning process of the granulations. One of the effects of the asymmetries is that, in general, the collision time is kept at a fraction of the idle time.

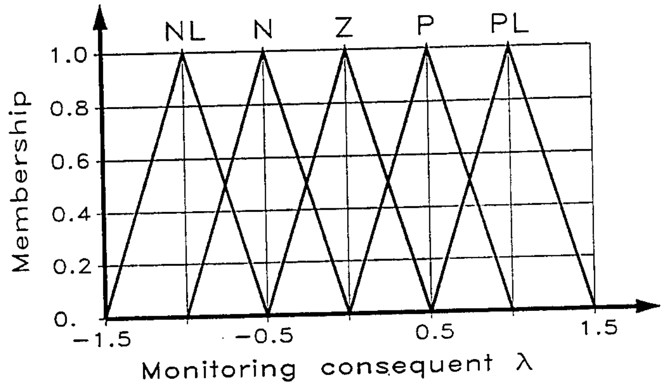

Figure 8. Granulation of the monitoring control'consequent λ.

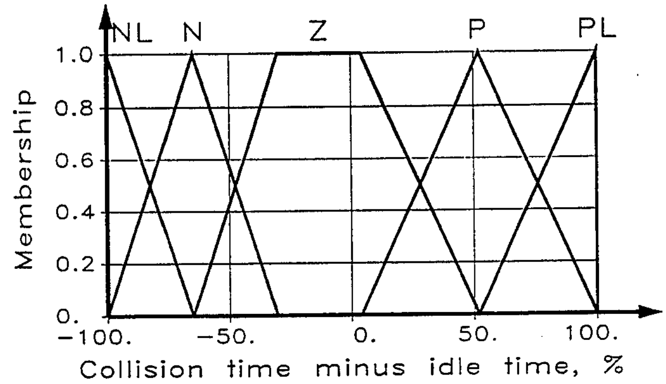

Figure 9. Granulation of the antecedent in Rule No. 1.

The granulations for the Rules No. 3 and 4 in Figures 11 and 12, respectively, are not normalized to the maximum value of unity. The consequence of such a granulation is that for input in the lower linguistic categories the effect (output) of the rule is small. However, during the development and tuning of the rules, we found that a better performance was obtained if the rule was not fired at all in the lower categories. Therefore, we formulated the rules correspondingly with "no output" for lower category input, see pages 11-12. For these rules, only the linguistic input categories P and PL have consequences, and the consequence of P is smaller than that of PL. The "thresholds" that are mentioned in the linguistic description of Rules 3 and 4 correspond to the input categories "zero" in Figures 11 and 12 and equal 60% for both rules.

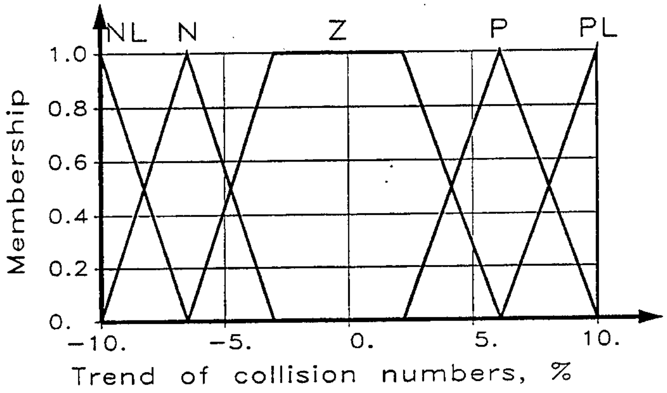

Figure 10. Granulation of the antecedent in Rule No. 2.

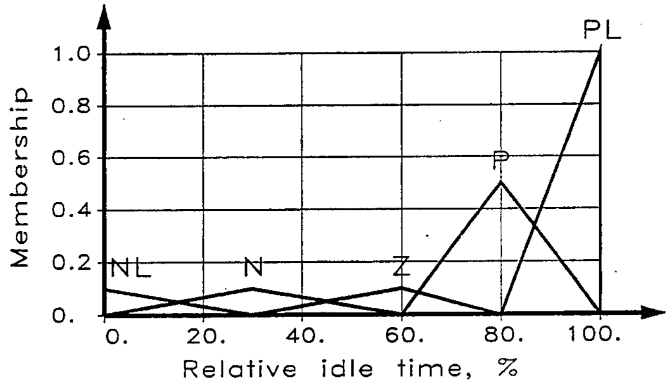

Figure 11. Granulation of the antecedent in Rule No. 3.

5. Examples

5.1. Determination of Optimal Granulations

The task of the presented network access control is to maintain a high information throughput rate in a congested network. (We call a network "congested" if several nodes have queues of messages ready to be broadcast and are competing for channel access time.) To compare the efficiency of different control strategies, one needs a measure for the information throughput rate. A simple and general measure is, for instance, the number of successfully transmitted bits per time unit, averaged over a convenient time interval. This would require, however, a knowledge of the transmission codes and the transmission rates that both are not relevant for the development of network access controls. For such controls, the only relevant property of a message is its length in seconds and, therefore, a network performance measure should be based on message lengths rather than on their contents or transmission rates in bits per second. We now give a description of such a

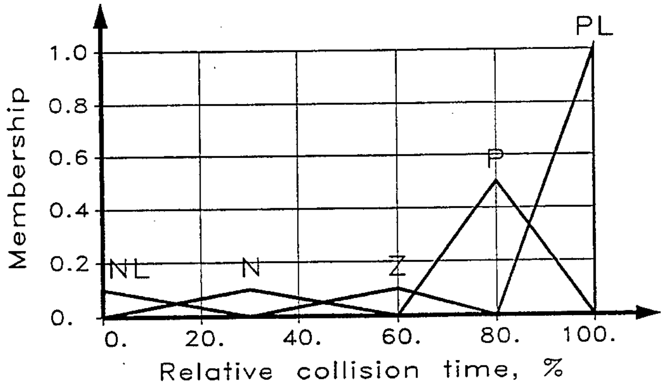

Figure 12. Granulation of the antecedent in Rule No. 4.

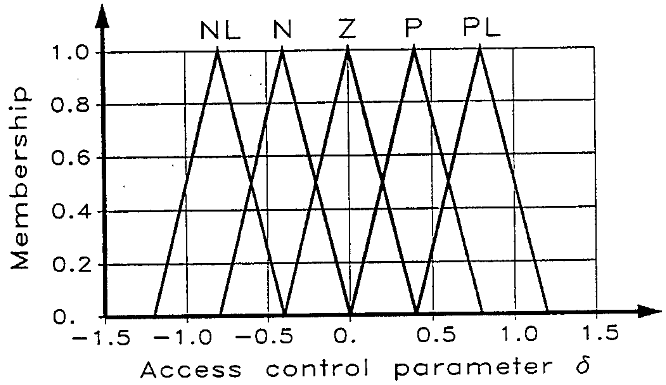

Figure 13. Granulation of the access control consequent δ.

measure.

In this study, congested networks were simulated by computer experiments as follows. (For details, see Celmiņš (1995).) To start an experiment, first a list of future messages was prepared for each node, whereby the number and generation times of the messages was chosen such that a congested condition would result. Next, a network clock was started and the nodes allowed to access the network and broadcast their messages in accordance with the rules of the current access control. The experiment was ended when all message queues had been cleared. For such experiments, a simple measure for the effectiveness of the access control is the end time of the experiment (i.e., the time needed to clear all queues). A control that consistently produces shorter clearance times is better then a control that causes longer clearance times.

Any individual access to the net depends on the random numbers that are drawn by the access managers at the nodes to set a time for the next broadcast (Section 2.2).

Therefore, the clearance time for a given collection of messages and a given control algorithm is not predictable but varies depending on the random number sequences that are used by the access managers. To obtain *typical* clearance time values, the computer simulations were repeated 20, 40, or more times with different seed numbers of the random number generator program and average clearance times were computed from these repetitions. These experiments indicated that, in general, 20 repetitions were sufficient to obtain typical and consistent results, but more cases had to be computed for fine-tuning of the granulations of the rules.

The use of the queue clearance time as an effectiveness measure has the disadvantage that the clearance time is dimensional and depends on the size of the network, the lengths of the messages, and the number of messages. This makes it difficult to compare performances among networks and among cases with different message lists. Therefore, the measure was modified. The modified measure, called the *slowness index*, was defined as the ratio of the queue clearance time to the theoretical minimum that is needed to transmit all the messages from the message lists. The theoretical minimum was computed by adding at the beginning of the experiment the lengths of all messages, including their expected acknowledgments, that were in the lists of future messages. The slowness index of the best control algorithms has typically a value of about two. The variation of the slowness index due to different random number sequences was found to be about plus or minus 0.25.

An example of the use of the slowness index for the tuning of the access controls is shown in Table 1. It contains the results of experiments with different granulations of the control Rules No. 1 and 2. The optimal granulations for these rules are shown in Figures 9 and 10, respectively, and Table 1 shows how deviations from optimal granulation affect the slowness index. The experiment involved a four-node network, and the values of the slowness index shown in the table are averages over 100 cases with different seed numbers. The rows correspond to the slowness index averages for different sizes of the core of the "zero" category membership function (the flat portion in Figure 9) of the antecedent of Rule No. 1. In this study, the granulation was symmetric and the half-width of the core was varied between zero and 100%. The columns correspond to granulations with different "zero" membership cores for the antecedent of Rule No. 2 (see Figure 10). Here again, the granulation was symmetric, but the half-width of the core was varied between zero and 10%. The minimum of the slowness index in this example is about 1.91 and its location is (2,20). Similar and more detailed experiments were carried out for a number of parameters of the granulations of all antecedents and consequents, as well as for other relevant parameters of the control algorithms. The optimal granulations were found to vary little over different conditions (different numbers of nodes and different characteristics of message lists). Also, the minima for the slowness index were found to be broad indicating that it is not necessary to tune the control rules very precisely. This means that the control is robust in the sense that it can be expected to perform well under quite

Table 1. Slowness index.

Antecedent of Rule 2.

		0.	2.	4.	6.	8.	10.
Antecedent of Rule 1.	0.	1.99	2.02	2.21	2.47	2.66	2.75
	20.	2.04	1.91	1.97	2.21	2.63	2.73
	40.	2.10	1.96	2.02	2.16	2.61	2.64
	60.	2.16	2.04	2.16	2.27	2.53	2.46
	80.	2.17	2.06	2.17	2.27	2.50	2.48
	100.	2.17	2.05	2.17	2.29	2.50	2.49

different conditions. The final optimal granulations are shown in Figures 7 through 13.

5.2. Examples of Control Performance

We first present some results of a computer experiment with a four-node congested network. In this example, during the first ten minutes of simulated time, the four nodes were generating messages at high rates so that eventually all nodes accumulated message queues. The message generation rates and the lengths of the messages were chosen such that all nodes generated in ten minutes approximately the same total amount of information (measured by the total length of all messages). After ten minutes the message generation was terminated, but the network remained active until all queues were cleared by about 28 minutes. Figure 14 shows the lengths of the message queues at the four nodes of the network. One notices that while the input was such that the total lengths of information were approximately equal for all nodes, some nodes cleared their queues faster than others. The reason for this is that the average lengths of the messages were different. Thus, Node No. 1 had few long messages, while Node No. 4 had a large number of short messages. The clearance times of the message queues show that in a noise-free network it is advantageous to combine messages into larger packets.

Figure 15 shows the cumulative network usage times during the activity of the network. It indicates that during the 28 minutes of activity about 16 minutes had been used to transmit messages, 3 minutes of network time were wasted with colliding messages, and 9 minutes was the total idle time. The slowness index was in this case 1.73. We recall that the access algorithm is such that one can reduce the idle time portion only by increasing the collision time and vice versa. Accordingly, Rule No. 1 of the access control (see Section 4.2.2, page 11) was formulated to adjust the access parameter (the access time delay interval D) such that the idle time and collision time are about equal. However, computer experiments show that the best (smallest) slowness index usually is obtained when the relative collision time is a fraction of the relative idle time. Such a suppression of the collision time is automatically achieved by the asymmetric granulations of the control rules.

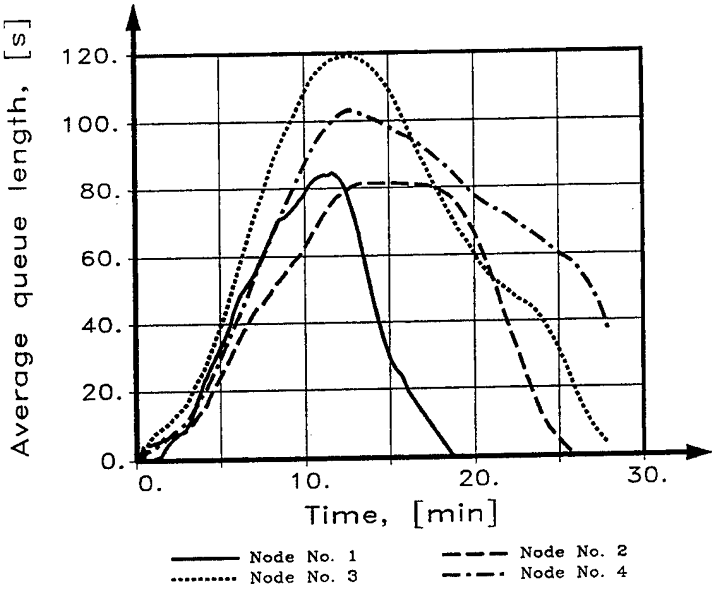

Figure 14. Message queues in a four-node network.

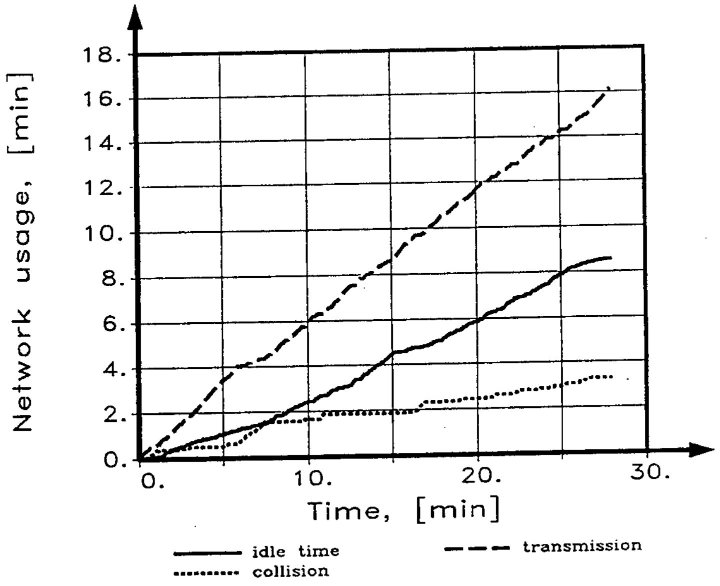

Figure 15. Time usage for queue clearing.

Figure 16 shows the relative network usage times. The data in this figure is the basic input for Rule No. 1. By that rule, the access time delay interval D is increased whenever the relative collision time increases. The obtained variation of D is illustrated in Figure 17. Comparing Figure 16 with Figure 17, one can observe that the control parameter D indeed generally increases when collision times increase. The correspondence is not perfect, because the other three rules interfere with the output of Rule No. 1. Toward the end of the experiment, there were fewer active nodes. Consequently, the number of collisions decreased and D was eventually reduced to near zero to allow a speedy clearance of the remaining queues.

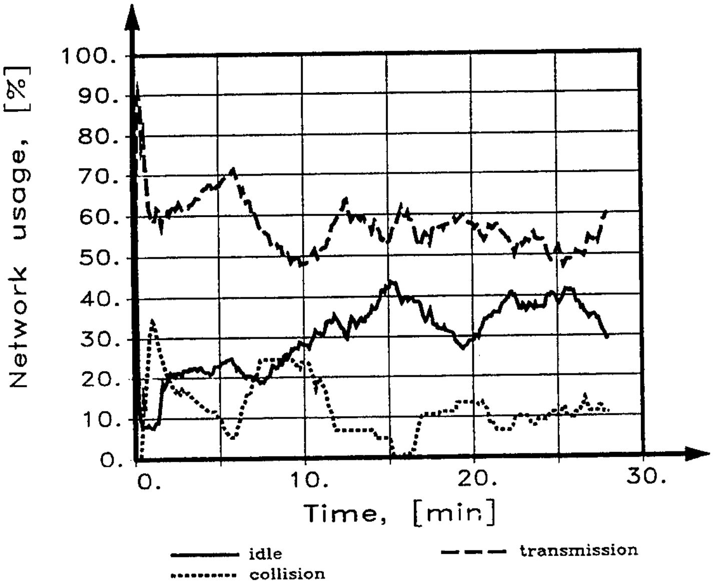

Figure 16. Monitored relative usage times.

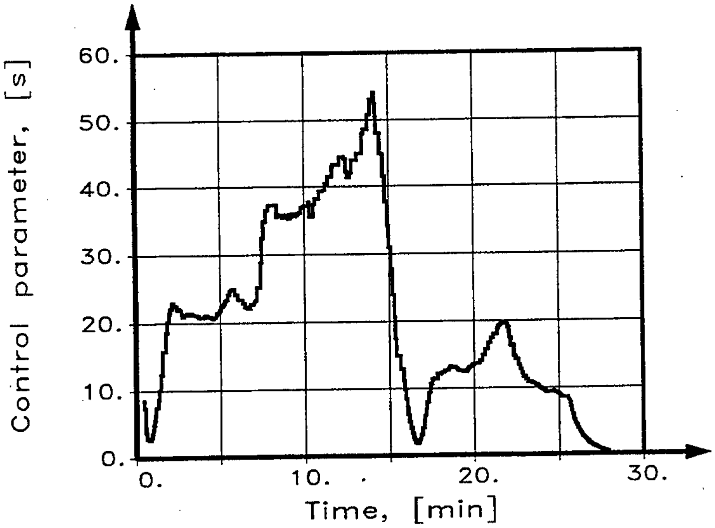

Figure 17. Variation of the delay time interval D.

The next example shows the performance of a not well-tuned controller. In this example, the network consisted of seven nodes and the monitoring interval L was fixed to 30 minutes. The excessive length of the monitoring interval caused a slow response to changing network conditions. Figure 18 shows the time usage during the clearance of the queues, and Figure 19 shows the variations of the access time delay interval D. We observe from Figure 18 that very few collisions were observed between 30 and 50 minutes simulated time, but Figure 19 shows that the control parameter D did not start to decrease before 47 minutes simulated time. A quicker response (due to a shorter monitoring interval) would reduce D sooner and save some of the idling time. The slowness index was 2.17 in this example. The average value of the slowness index for this type of network and message queues is about 1.91 when optimal control is used.

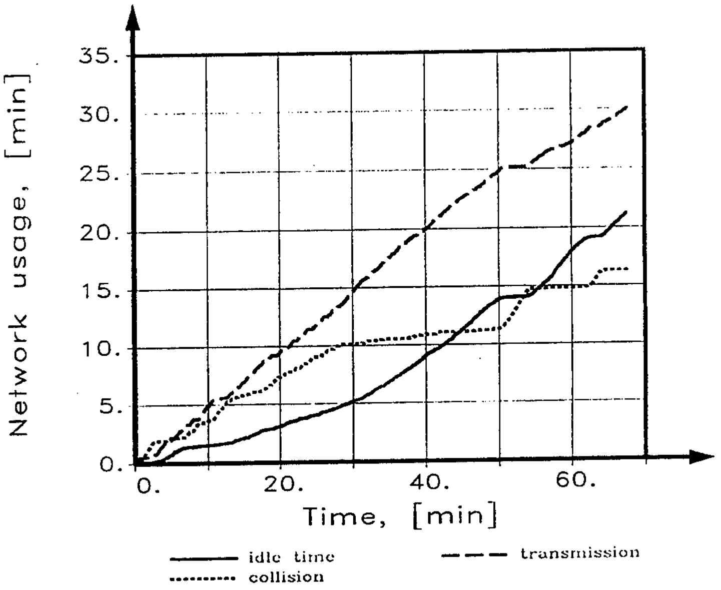

Figure 18. Time usage with a badly tuned control.

We note that these examples are not representative but only illustrative. Because the access manager uses random access times and any broadcast influences all subsequent transmissions, details of experiments with different random number sequences can be quite different even for identical message lists and identical control algorithms. Optimality of the control algorithm was sought and determined only in the average for a large number of experiments.

6. Summary and Conclusions

This report describes the principles of a distributed control procedure for battlefield communications. The control procedure is designed such that each participant (node) in the network controls its own access based on the perceived network conditions. Because the controlling agents are distributed among all nodes, the controlling system is virtually invulnerable. The control is based on fuzzy logic, and the control parameters were determined with the help of computer experiments such that the information throughput rate of the network is maximized in the average over prolonged operations.

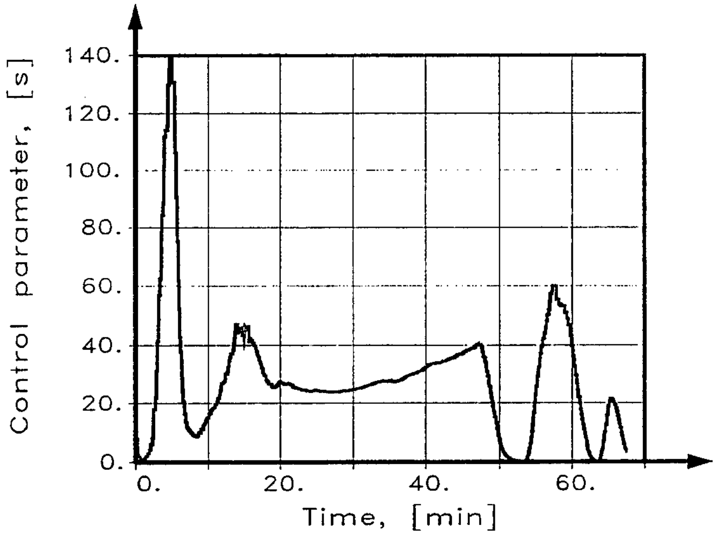

Figure 19. Variation of *D* by a badly tuned control.

The control algorithms that are developed in this report are sufficiently simple for an implementation by software on PC-type computers. The software can be developed inhouse or commercial fuzzy-logic control tools might be used. A discussion of commercial tools can be found, for instance, in Yen, Langari, and Zadeh (1995).

An alternative approach is to design fuzzy computer chips specifically for the described communication control purposes. Tools for the development of fuzzy chips are offered by several companies, and the cost of the chips is small. Examples of suppliers of dedicated fuzzy-logic chips are Togai Infralogic, Adaptive Logic, and ICCT Technologies. The advantages of a dedicated hardware are a faster response of the controller, and a more robust and tamper-resistant product.

INTENTIONALLY LEFT BLANK

7. References

Celmiņš, Aivars, Battlefield Communications Network Model (BATNET). ARL-MR-244, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, August 1995.

Kaste, Virginia A. T., Ann M. Brodeen, and Barbara D. Broome, An Experiment to Examine Protocol Performance Over Combat Net Radios. BRL-MR-3979, U.S. Army Ballistic Research Laboratory. Aberdeen Proving Ground, MD, June 1992.

Klir, George J. and Bo Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications,* Prentice Hall PTR, Upper Saddle River, NJ, 1995.

Pedrycz, Witold, *Fuzzy Control and Fuzzy Systems,* 2nd Extended Edition, John Wiley & Sons, New York, 1992.

Terano, Toshiro, Kiyoji Asai, and Michio Sugeno, *Fuzzy Systems Theory and Its Applications,* Academic Press, San Diego, CA, 1991.

Yen, John, Reza Langari and Lotfi A. Zadeh, *Industrial Applications of Fuzzy Logic and Intelligent Systems,* IEEE Press, New York, 1995

Zadeh, Lotfi A., The Concept of a Linguistic Variable and its Application to Approximate Reasoning, Part III, *Information Sciences* 9, pp. 43-80, 1975.

INTENTIONALLY LEFT BLANK

NO. OF
COPIES ORGANIZATION

2 ADMINISTRATOR
ATTN DTIC DDA
DEFENSE TECHNICAL INFO CTR
CAMERON STATION
ALEXANDRIA VA 22304-6145

1 DIRECTOR
ATTN AMSRL OP SD TA
US ARMY RESEARCH LAB
2800 POWDER MILL RD
ADELPHI MD 20783-1145

3 DIRECTOR
ATTN AMSRL OP SD TL
US ARMY RESEARCH LAB
2800 POWDER MILL RD
ADELPHI MD 20783-1145

1 DIRECTOR
ATTN AMSRL OP SD TP
US ARMY RESEARCH LAB
2800 POWDER MILL RD
ADELPHI MD 20783-1145

ABERDEEN PROVING GROUND

2 DIR USARL
ATTN AMSRL OP AP L (305)

NO. OF
COPIES ORGANIZATION

1 HQDA
ATTN SAUS OR MR W HOLLIS
PENTAGON
WASHINGTON DC 20310-0001

1 SUPERINTENDANT
US MILITARY ACADEMY
ATTN MADN A
WEST POINT NY 10996-1786

2 US MILITARY ACADEMY
ATTN MAJ DON ENGEN
CPT ROD STURDIVANT
DEPT OF MATH SCI
WEST POINT NY 10996-1786

3 PROJECT MANAGER FATDS
MR FRANK MURPHY
MR MICHAEL LYONS
MR HENRY SAPHOW
BLDG 457
FT MONMOUTH NJ 07703-5027

17 PM FATDS
ATTN AMCPM TF A LTC DIXON
BLDG 457
FT MONMOUTH NJ 07703-5027

2 DIRECTOR
US ARMY RESEARCH LABORATORY
ATTN AMSRL SS IC
MR EMMERMAN
MR TOKARCIK
2800 POWDER MILL ROAD
ADELPHI MD 20783-1197

2 DIRECTOR
US ARMY ASCO
ATTN AMSLC AT
BLDG 2424
FT MEADE MD 20755-5313

3 COMMANDING GENERAL
USMC RSRCH DEV AND ACQ CMD
ATTN EXECUTIVE DPM
COL STANKOWSKY
QUANTICO VA 22134-5000

NO. OF
COPIES ORGANIZATION

3 DIRECTOR
US ARMY RESEARCH OFFICE
ATTN SLCRO EL DR GAULT
PO BOX 12211
RESEARCH TRIANGLE PARK NC
27709-2211

1 US ARMY RESEARCH OFFICE
ATTN DR R R LAUNER
MATH AND COMP DIV
PO BOX 12211
RESEARCH TRIANGLE PARK NC
27709-2211

1 PROJECT MANAGER ADCCS
ATTN SFAE CC AD MR S ALLEN
4920 UNIVERSITY SQUARE ST 3B
HUNTSVILLE AL 35816

2 PROJECT MANAGER FAAD C2
ATTN SPIC CC ADCCS FA
REDSTONE ARSENAL AL
35098-5600

2 PROJECT MANAGER ASAS
ATTN JTFDO ASAS
JTFDO AE C
1500 PLANNING RESEARCH DR
MCLEAN VA 22102

3 PROJECT MANAGER CHS
ATTN SSAE CC CHS
COL OLSON
MR LEVINE
MR BYRNILDSEN
FT MONMOUTH NJ 07703-5000

1 PROJECT MANAGER OPTADS
ATTN SPIC CC OTDS
FT MONMOUTH NJ 07703-5000

4 PEO CMND AND CONTROL SYS
ATTN SFAE CC
MR HARMON
MR GIORDANO
MR ALBARELLI
SFAE CC SE MR MERCURIO
FT MONMOUTH NJ 07703-5000

NO. OF COPIES	ORGANIZATION
5	COMMANDER US ARMY CECOM ATTN AMSEL RD AED MR RUTH AMSEL RD C3 CC DR KLOSE MR QUIGLEY MR STROZYK AMSEL RD C3 TP E MR GRAFF FT MONMOUTH NJ 07703-5001
3	COMMANDER US ARMY ARDEC ATTN TECH DIR DR DAVIDSON SMCAR FSF MR LEHMAN SMCAR ASH MR LARRY OSTUNI PICATINNY ARSENAL NJ 07806-5000
1	COMMANDER US ARMY MATERIEL COMMAND ATTN AMCDRA DR LUCY HAGAN 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333-0001
2	COMMANDANT US ARMY FIELD ARTILLERY SCHOOL ATTN ATSF TSM C3 COL COOPER FT SILL OK 73503-5000
4	COMMANDANT US ARMY FIELD ARTILLERY SCHOOL ATTN ATSF FSC3 MAJ SELLS CPT WILLIAMS MAJ SCOTT CPT BLAKLEY FT SILL OK 73501
2	COMMANDER USMC RSRCH DEV & ENGRG CMND ATTN PM GROUND C2 QUANTICO VA 22134-5080
1	NATL INST OF STNDRDZTN AND TECH ATTN MR SANDOR SZABO BLDG 220 RM B127 GAITHERSBURG MD 20899
1	DIRECTOR SANDIA NATIONAL LABS ATTN DR LARRY CHOATE PO BOX 5800 ALBUQUERQUE NM 87185
1	COMMANDER USACADA ATTN TPIO FT LEAVENWORTH KS 66027-5000
2	COMMANDANT US ARMY FIELD ARTILLERY SCHOOL ATTN ATSF CD DIR OF COMBAT DEV FT SILL OK 73503-5000
1	COMMANDANT US ARMY FIELD ARTILLERY SCHOOL ATTN ATSF CD MR DUBLISKY FT SILL OK 73503-5000
1	COMMANDANT US ARMY FIELD ARTILLERY SCHOOL ATTN PRESIDENT FA BD COL ELDER FT SILL OK 73503-5000
6	MAGNAVOX ATTN TOM KLAGE 4 CP MIKE MEIER 10 06 2 CP 1313 PRODUCTION ROAD FT WAYNE IN 46808
2	COMMAND SYSTEMS INC ATTN DEVIN R WILLIS MARK MCCLEARY 1025 GOSHEN ROAD FT WAYNE IN 46808
2	PM SINCGARS ATTN SFAE CM GAR COL DOMINIC F BASILE FT MONMOUTH NJ 07703-5000
1	PM ADDS ATTN SFAE CM ADD COL LELAND H HEWITT FT MONMOUTH NJ 07703-5000

NO. OF
COPIES ORGANIZATION

1 PM EPLRS
ATTN SFAE CM ADD EPL
LTC C FORNECKER
FT MONMOUTH NJ 07703-5000

1 PM MSE
ATTN SFAE CM MSE
COL DAVID GUST
FT MONMOUTH NJ 07703-5000

2 COMMANDANT
USAFAS
ATTN ATSF CCS
WALTER W MILLSPAUGH
MAJ DRUMMUND
FT SILL OK 73503-5600

4 OPM USAFAS
ATTN SFAE ASM FA
LARRY YUNG
PICATINNY ARSENAL NJ
07806-5000

1 COMMANDER
US ARMY MICOM
ATTN AMSMI RD SED
MR G CLAYTON
REDSTONE ARSENAL AL
35898-5620

2 COMMANDER
US ARMY TACOM
ATTN AMSTA RV
MR SARNA
MR HALLE
WARREN MI 48397-5000

3 COMMANDER
USCAC
ATTN ATZL CDC
COL BAERMAN
LTC COOK
CPT BROWN
FT LEAVENWORTH KS
66027-5300

NO. OF
COPIES ORGANIZATION

4 DIRECTOR
US ARMY RESEARCH LAB
ATTN AMSRL SC I
DR GANTT
MR MITCHELL
MR RACINE
COL BLAKE
115 O KEEFE BLDG GIT
ATLANTA GA 30332-0800

1 BOEING COMPUTER SERVICES
ATTN DR OSCAR KIPERSZTOK
RESEARCH AND TECHNOLOGY
MS 7L 41
PO BOX 24346
SEATTLE WA 98124-0346

2 UNIVERSITY OF MARYLAND
ATTN PROF BILAL M AYYUB
DR OSCAR CHANG
DEPT OF CIVIL ENGNRG
COLLEGE PARK MD 20742

1 UNIVERSITY OF DELAWARE
ATTN PROF SHINYA KIKUCHI
DEPT OF CIVIL ENGNRG
342 DUPONT HALL
NEWARK DE 19716

1 JOHNS HOPKINS UNIVERSITY
ATTN DR RAMASWAMY MURALI
105 BARTON HALL
BALTIMORE MD 21218

1 HIGH PERFORMANCE TECH INC
ATTN MR ROBERT POST
11417 SUNSET HILLS ROAD
SUITE 106
RESTON VA 22090

1 ARIZONA STATE UNIVERSITY
ATTN PROF SUMIT GHOSH
DEPT OF CMPTR SCI AND ENGNRG
BOX 875406
TEMPE AZ 85287-5406

NO. OF
COPIES ORGANIZATION

ABERDEEN PROVING GROUND

9 DIR USAMSAA
ATTN AMXSY C
MR HAL BURKE
MR PETE REID
MR TOM NOLAN
MR PAT WARD
MR FRANK FOX
MR JOH DILEO
AMXSY G MR JOHN KRAMER
AMXSY A MR WALTER CLIFFORD
JTCG ME MR ART LAGRANGE

31 DIR USARL
ATTN AMSRL WT WF
G HORLEY
W DOUSA
A THOMPSON
AMSRL HR SA D TYROL
AMSRL IS TP
A BRODEEN
B BROOME
S CHAMBERLAIN
B COOPER
A DOWNS
G HARTWIG
P KASTE
M MARKOWSKI
AMSRL SC C H BREAUX
AMSRL SC CC
C NIETUBICZ
D PRESSEL
M TAYLOR
C ZOLTANI
D HISLEY
J GROSH
J COLLINS
B BODT
A CELMINS (3 CP)
AMSRL SC I
T HANRATTY
J DOMER
R HAMMELL
R HELFMAN
AMSRL SC S V KASTE
AMSRL SC SS M THOMAS
AMSRL SL C W HUGHES

INTENTIONALLY LEFT BLANK.

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1215 (Celmiņš) Date of Report October 1996

2. Date Report Received

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.)

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate.

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

CURRENT ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)